The following properties are among the most important:  
  
 In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code..  
  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Use of a static code analysis tool can help detect some possible problems.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input.  
However, readability is more than just programming style.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
 Code-breaking algorithms have also existed for centuries.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
Ideally, the programming language best suited for the task at hand will be selected.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.